

Computing Sentiment Polarity of Texts at Document and Aspect Levels

Vivek Kumar Singh¹, Rajesh Piryani²,
Pranav Waila³, and Madhavi Devaraj⁴, Non-members

ABSTRACT

This paper presents our experimental work on two aspects of sentiment analysis. First, we evaluate the performance of different machine learning as well as lexicon based methods for sentiment analysis of texts obtained from variety of sources. Our performance evaluation results are on six different datasets of different kinds, including movie reviews, blog posts and twitter feeds. To the best of our knowledge no such work on comprehensive evaluative account involving different techniques on variety of datasets have been reported earlier. The second major work that we report here is about the heuristic based scheme that we devised for aspect-level sentiment profile generation of movies. Our algorithmic formulation parses the user reviews for a movie and generates a sentiment polarity profile of the movie based on opinion expressed on various aspects in the user reviews. The results obtained for the aspect-level computation are also compared with the corresponding results obtained from the document-level approach. In summary, the paper makes two important contributions: (a) it presents a detailed evaluative account of both supervised and unsupervised algorithmic formulations on six datasets of different varieties, and (b) it proposes a new heuristic based aspect-level sentiment computation approach for movie reviews, which results in a more focused and useful sentiment profile for the movies.

Keywords: Aspect-oriented Sentiment, Document-level Sentiment, Opinion Mining, Sentiment Analysis, SentiWordNet.

1. INTRODUCTION

Sentiment analysis is language processing task that uses an algorithmic formulation to identify opinionated content and categorize it as having ‘positive’, ‘negative’ or ‘neutral’ polarity. It has been formally

defined as an approach that works on a quintuple $\langle O_i, F_{ij}, Sk_{ijl}, H_k, T_l \rangle$; where, O_i is the target object, F_{ij} is a feature of the object O_i , Sk_{ijl} is the sentiment polarity (+ve, -ve or neutral) of opinion of holder k on j th feature of object i at time l , and T_l is the time when the opinion is expressed [1]. It can be clearly inferred from this definition that sentiment analysis involves a number of tasks ranging from identifying whether the target carries an opinion or not and if it carries an opinion then to classify the opinion as having ‘positive’ or ‘negative’ polarity. In this paper, we have restricted our discussion to computing sentiment polarity only and have purposefully excluded the subjectivity analysis.

The sentiment analysis task may be done at different levels, document-level, sentence-level or aspect-level. The document-level sentiment analysis problem is essentially as follows: given a set of documents D , a sentiment analysis algorithm classifies each document $d \in D$ into one of the two classes ‘positive’ or ‘negative’. Positive label denotes that the document d expresses an overall positive opinion and negative label means that d expresses an overall negative opinion of the user. Sometimes degree of positivity or negativity is also computed. The document-level sentiment analysis assumes that each document contains opinion of user about a single object. If the document contains opinions about multiple objects within the same document, the sentiment analysis results may be inaccurate. The aspect-level sentiment analysis on the other hand assumes that a document contains opinion about multiple aspects/ entities of one or more objects in a document. It is therefore necessary to identify about which entity an opinion is directed at. The phrases and sentence structures are usually parsed by using knowledge of linguistics for this purpose.

There are broadly two kinds of approaches for sentiment analysis: those based on machine learning classifiers and those based on lexicon. The machine learning classifiers for sentiment analysis are usually a kind of supervised machine learning paradigm that uses training on labelled data before they can be applied to the actual sentiment classification task. In the past, varieties of machine learning classifiers have been used for sentiment analysis, such as naive bayes, support vector machine and maximum entropy classifiers. The lexicon-based methods usually employ a

Manuscript received on August 31, 2013.

Final manuscript received March 31, 2013.

^{1,2} The authors are with Department of Computer Science, South Asian University, New Delhi, India, E-mail: vivek@cs.sau.ac.in and rajesh.pirayani@gmail.com

³ The author is with DST-CIMS, Banaras Hindu University, Varanasi, India, E-mail: pranav.waila@gmail.com

⁴ The author is with Department of Computer Science & Engineering, Gautam Buddha Technical University, Lucknow, India, E-mail: madhavidavaraj@gmail.com

sentiment dictionary for computing sentiment polarity of a text. Another kind of effort for sentiment analysis includes the SO-PMI-IR algorithm, which is a purely unsupervised approach, which uses the mutual occurrence frequency of selected words on the World Wide Web (hereafter referred to as Web) in order to compute the sentiment polarity.

Sentiment analysis is now a very useful task across a wide variety of domains. Whether it is commercial exploitation by organizations for identifying customer attitudes/ opinions about products/ services, or identifying the election prospect of political candidates; sentiment analysis finds its applications. The user created information is of immense potential to companies which try to know the feedback about their products or services. This feedback helps them in taking informed decisions. In addition to being useful for companies, the reviews are helpful for general users as well. For example, reviews about hotels in a city may help a user visiting that city in locating a good hotel. Similarly, movie reviews help other users in deciding whether the movie is worth watch or not. However, the large number of reviews becomes information overload in absence of automated methods for computing their sentiment polarities. Sentiment analysis fills this gap by producing a sentiment profile from large number of user reviews about a product or service. The new transformed user-centric, participative Web allows extremely large number of users to express themselves on the Web about virtually endless topics. People now write reviews about movies, products, services; write blogs to express their opinion about different socio-political events; and express their immediate emotions in short texts on Twitter. The social media is now a major phenomenon on the Web and a large volume of the content so created is unstructured textual data. It is this reason why sentiment analysis has become an important task in text analytics with lots of applications.

The rest of the paper is organized as follows. Section 2 describes the popularly used document-level sentiment analysis approaches based on machine learning classifiers (such as naive bayes and support vector machine) and algorithmic formulation based on the SentiWordNet library. The section 3 describes the dataset used, performance metrics computed and experimental setup for document-level sentiment analysis task. Section 4 presents the experimental results of document-level sentiment analysis on different datasets. The section 5 describes our algorithmic design for aspect-level sentiment profiling and the corresponding experimental results. The paper concludes with a summary of the contribution of this work stated in Section 6.

2. DOCUMENT-LEVEL SENTIMENT ANALYSIS

Sentiment analysis at document-level has been explored in many past works. Pang Lee et al. in their work reported in 2002 [2] and 2004 [3] applied naive bayes, support vector machine and maximum entropy classifiers for document-level sentiment analysis of movie reviews. In a later work reported in 2005 [4], they have applied support vector machine, regression and metric labeling for assigning sentiment of a document using a 3 or 4-point scale. Gamon in a published work in 2004 [5] used support vector machine to assign sentiment of a document using a 4-point scale. Dave et al. in their work reported in 2003 [6] used scoring, smoothing, naive bayes, maximum entropy and support vector machine for assigning sentiment to documents. Kim and Hovy in a work reported in 2004 [7] used a probabilistic method for assigning sentiment to expressions. Turney in the work reported in 2002 [8], [9] used the unsupervised SO-PMI-IR algorithm for sentiment analysis of movie and travel reviews. Bikel et al. in their work in 2007 [10] implemented subsequence kernel based voted perceptron and compared its performance with standard support vector machine. Durant and Smith [11] tried sentiment analysis of political weblogs; Sebastiani [12] and Esuli & Sebastiani [13] worked towards gloss analysis and proposed the SentiWordNet approach for sentiment analysis. Many other important works have been reported on sentiment analysis at document-level. However, we did not aim to present a detailed survey on sentiment analysis, which can be found in some recent works [14] and [15]. Here, our aim is largely to compare the performance of machine learning classifier based approaches vis-à-vis unsupervised SentiWordNet based approaches for sentiment analysis of diverse textual data. We briefly describe the three important approaches we compared in the following paragraphs.

2.1 Naive Bayes Algorithm

It is supervised probabilistic machine learning classifier that can be used to classify textual documents. The sentiment analysis problem using naive bayes (NB) classifier can be visualized as text classification problem of two classes, those with 'positive' polarity and those with 'negative' polarity. Every document is thus assigned to one of these two classes. The main concern that needs to be addressed while using naive bayes classifier for sentiment analysis is whether all terms occurring in documents should be used as features as it done in normal text classification or to select specific terms which may be in more concrete forms of expression of opinions. We have used full term profile based on result reported in [2] and [3] that accuracy of classification improves if all frequently occurring words are considered rather than only adjectives. Once the feature selection is done,

the actual classification task is a simple probabilistic estimate based on term occurrence profiles. In naive bayes classifier, the probability of document d being in class c is computed as:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (1)$$

where, the term $P(c)$ refer to prior probability of a document occurring in class c and corresponds to the majority class. The expression $P(t_k/c)$ is the conditional probability of a term t_k occurring in a document of class c . The term $P(t_k/c)$ is interpreted to be the measure of how much evidence the term t_k contributes that c is correct class. The main idea in this classification is to classify the document based on statistical pattern of occurrence of terms. The objective in text classification using naive bayes is to determine the best class for a document. The best class in naive bayes classification is the maximum posteriori (MAP) class which can be computed as:

$$c_{map} = \arg \max_{c \in C} P(c|d) = \arg \max_{c \in C} P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (2)$$

where, P indicates estimated value found from the training set. The multiplication of many conditional probability terms can be reduced by adding logarithms of probabilities. Therefore, equation (2) can be written as:

$$c_{map} = \arg \max_{c \in C} \left[\log P(c) + \sum_{1 \leq k \leq n_d} \log P(t_k|c) \right] \quad (3)$$

In equation (3) each $P(t_k/c)$ term refer to the weight which specify how good an indicator the term t_k is for class c , and in similar way the prior $\log P(c)$ indicates the relative frequency of class c . The naive bayes approach has two variants: the multinomial naive bayes and the Bernoulli's naive bayes [16]. We have implemented the multinomial naive bayes which takes into account the term occurrence frequencies as opposed to only measuring term presence in the Bernoulli's naive bayes scheme. We have used only Unigram results for comparison of different methods across the datasets.

2.2 Support Vector Machine Algorithm

Support Vector Machine (SVM) is another well-known and wide margin machine learning based classifier. It is vector space model based classifier that needs feature vectors transformed into numerical values before it can be used for classification. Usually the text documents are converted to a multidimensional *tf.idf* vector. Now, the whole problem is to classify each text document represented by the feature vectors in specific classes. Here, the main idea

is to find a decision vector/surface that is maximally away from any data point (document vectors in our case). The margin of the classifier can be found out through distance from the decision surface to the closest data point. The target is to maximize this margin. Suppose $D = \{x_i, y_i\}$ represents the training set data points, where each element refers to pair of point x_i and a class label y_i corresponding to it. The two data classes are always named as +1 and -1 and support vector machine as such is a linear classifier two-class classifier [16]. Then linear classifier is:

$$f(x) = \text{sign}(w^T x + b) \quad (4)$$

The value of -1 represent one class and value of +1 represents the other class. The reduction problem that attempts to determine w and b such that (a) $\frac{1}{2}w^T w$ is minimized, and (b) for all $\{(x_i, y_i)\}$, $y_i(w^T x_i + b) \geq 1$. This represents the quadratic optimization problem that can be solved by means of standard quadratic programming libraries. In the solution, a lagrange multiplier α_i is related with each constant $y_i(w^T x_i + b) \geq 1$. The goal is then to find $\alpha_1, \alpha_2, \dots, \alpha_N$ such that:

$$\sum_{\alpha_i} -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (5)$$

is maximized subject to constraints $\sum_i \alpha_i y_i = 0$ and $\alpha_i \geq 0$ for all $1 \leq i \leq N$. The solution is of the form:

$$\begin{aligned} w &= \sum \alpha_i y_i x_i \\ b &= y_k - w^T x_k \end{aligned} \quad \text{for any such } x_k \text{ s.t. } \alpha_k \neq 0 \quad (6)$$

The classification function thus becomes:

$$f(x) = \text{sign} \left(\sum \alpha_i y_i x_i^T x + b \right) \quad (7)$$

In the solution, most of the α_i are zero. Each non-zero α_i indicates that the corresponding x_i is a support vector. In our experimental implementation, for solving the quadratic optimization problem, we used Sequential Minimal Optimizer (SMO) available in weka[17]. Through SMO the quadratic programming problem splits into numerous small problems, solving these problems sequentially gives the same answer as solving the big quadratic convex problem. The support vector machine is used for sentiment analysis at document-level as it is essentially a two-class linear classifier. For feature vectors, we have used the entire vocabulary of the documents without any bias for selected words such as those having POS tag as adjectives.

2.3 SentiWordNet based Approach

The third algorithmic approach we implemented is an unsupervised lexicon based method based on the SentiWordNet library [12], [13]. A sentiment analysis

approach using this library parses the term profile of a textual review document, extracts terms having desired POS tags, compute their sentiment orientation values from the library and then aggregates all such values to assign either ‘positive’ or ‘negative’ label to the whole document. These approaches usually target terms with a specific POS tag, which are believed to be opinion carriers (such as adjectives, adverbs or verbs). Thus subjecting the review text to a POS tagger becomes a prerequisite step for applying the SentiWordNet based approaches. After doing POS tagging, words with appropriate POS tag labels are selected and the SentiWordNet library is looked into for their sentiment polarity scores. Usually the terms having ‘positive’ sentiment orientation, have polarity values greater than zero. Terms having ‘negative’ sentiment orientation, have polarity value less than zero. In the past, researchers have explored using words with POS tags adjectives, adjectives preceded by adverbs and verbs etc. The polarity scores for all extracted terms in a review document are then aggregated using some aggregation formula and the resultant score is used to decide whether the document should be labeled as having ‘positive’ or ‘negative’ sentiment. Thus, two key issues in SentiWordNet based approaches are to decide: (a) which POS tag patterns from the document should be extracted for lookup in the library, and (b) how to decide the weightage of scores of different POS tags extracted while computing the aggregate score.

We have implemented several versions of SentiWordNet based approach by exploring with different linguistic features and weightage & aggregation schemes. Studies in computational linguistics suggest that adjectives are good markers of opinions. For example, if a review sentence says “The movie was excellent”, then use of adjective ‘excellent’ tells us that the movie was liked by the review writer and possibly s/he had a wonderful experience watching it. Sometimes, adverbs further modify the opinion expressed in review sentences. For example, the sentence “The movie was extremely good” expresses a more positive opinion about the movie than the sentence “the movie was good”. A related previous work [18] has also concluded that ‘adverb+adjective’ combine produces better results than using adjectives alone. Hence we preferred the ‘adverb+adjective’ combine over extracting ‘adjective’ alone. The adverbs are usually used as complements or modifiers. Few more examples of adverb usage are: *he ran quickly, only adults, very dangerous trip, very nicely, rarely bad, rarely good etc.* In all these examples adverbs modify the adjectives. Though adverbs are of various kinds, but for sentiment classification only adjectives of degree seem useful. Some other previous works on lexicon-based sentiment analysis reported in [19] and [20] state that including words with ‘verb’ POS tag plays a role in improving the sentiment classification accu-

racy. We have therefore implemented another version that incorporates verb scores as well. In total we implemented three variants of the SentiWordNet based sentiment analysis approach. The detailed implementation these schemes is explained in section 3.4.

3. DATASET AND EXPERIMENTAL SETUP

We evaluated performance of naïve bayes, support vector machine and SentiWordNet based sentiment analysis approaches on six different data sets.

3.1 Datasets

We used a total of six *datasets* for evaluating different sentiment analysis schemes. This includes two existing movie review data sets, one movie review *dataset* collected by us, two existing blog *datasets* and one twitter *dataset*. The existing movie review *datasets* are from Cornell sentiment polarity *dataset* [21]. We downloaded polarity *Dataset v1.0* (referred as *Dataset 1*) and *v2.0* (referred as *Datasets 2*). The *datasets 1* comprises of 700 positive and 700 negative processed reviews, whereas the *Dataset2* comprises of 1000 positive and 1000 negative processed reviews. The third *datasets* (referred as *Dataset 3*) is our own collection comprising of 1000 reviews of Hindi movies, with 10 reviews each of 100 Hindi movies from the movie database site IMDB [22]. The blog *datasets* are drawn from an earlier collection [23] and then processed and labeled using Alchemy API [24]. The blog data used is about the ‘Arab Spring’ is opinionated in nature. We refer these *datasets* as *Dataset 4* and *Dataset 5*. The Twitter *dataset* is obtained from [25] and is comprised of twitter feeds used earlier for sentiment analysis. We refer to this *dataset* as *Dataset 6*. Thus in total we work on three different kinds of data items, reviews, blog posts and twitter feeds. Some statistics about the *datasets* used is described in table 1.

Table 1: *Datasets.*

<i>Dataset Description</i>	<i>Size/ Number</i>	<i>Avg. length (in words)</i>
700+700 Movie Reviews	1400	655
1000+1000 Movie Reviews	2000	656
1000 Reviews of Hindi Movies	1000	323
Blog posts on Libyan Revolution	1486	1130
Blog posts on Tunisian Revolution	807	1171
Twitter Dataset	20000	13

3.2 Implementing Nave Bayes Algorithm

We have implemented the multinomial version of naïve bayes algorithm using JAVA with Eclipse IDE. All the labeled datasets have been fed to the Nave bayes algorithm as k-folds; in our case k is 10. A 10-fold application of test data means that the dataset is divided into 10 equal parts and then 9 of the 10 parts becomes the training data and remaining 1 part constitute test data. This is done by choosing each of the possible permutations as training and test data in different runs. We have taken the entire set of terms as features, both due to motivation from the past results and in order to compare the results with the ‘adverb+adjective’ and ‘adverb+verb’ combinations of SentiWordNet approach implementation. Average of the 10-fold runs is reported as the performance level.

3.3 Implementing Support Vector Machine Algorithm

The support vector machine (SVM) algorithm is implemented in the Weka environment. Being a vector space model based classifier; it first required us to transform the textual movie reviews to vector space representation. We used *tf.idf* representation for transforming the textual reviews to numerical vectors for all the six datasets. No stop word removal or stemming was performed. This was done purposefully so that no feature having sentimental value gets excluded in the representation. We have thereafter used the same fold scheme as stated earlier and run our implementation of SVM and observed the results. The reported results are averaged over 10-folds of runs.

3.4 Implementing SentiWordNet based Approaches

We here implemented three different versions of SentiWordNet based approach, all in Java using NetBeans 7 IDE. In the first implementation we used only ‘adjectives’ as features. However, we have used it only as a baseline for internal evaluation of the other two implementations and did not show its results in result tables. Since it has been reported in the past works [19], [20], that adverbs and verbs play an important role in accurate sentiment analysis, we implemented two more versions, for which we have shown the performance evaluation results. In the second version, we used both ‘adverbs’ and ‘adjectives’ as features. And in the third version, we used ‘adverb+adjective’ and ‘adverb+verb’ as features for sentiment polarity computation. Thus, in the second version implementation, we only extract ‘adjectives’ and any ‘adverbs’ preceding the adjectives. In the third version, we extract both ‘adjectives’ and ‘verbs’, along with any ‘adverbs’ preceding them.

Since ‘adverbs’ are modifying the scores of succeeding terms (in both the implemented versions), it

needs to be decided as to what proportion the sentiment score of an ‘adjective’ or a ‘verb’ should be modified by the preceding ‘adverb’. We have taken the modifying weightage (scaling factor) of adverb score as 0.35, based on the conclusions reported in [19]. The other main issue that remains to be addressed is how should the sentiment scores of extracted ‘adverb+adjective’ and ‘adverb+verb’ combines in a sentence of the review document should be aggregated. For this we have tried different weight factors ranging from 10% to 100%, i.e. the ‘adverb+verb’ scores are added to ‘adverb+adjective’ scores in a weighted manner, giving weightage of 10-100% in the aggregated score. Thus if sentiment polarity score total of an ‘adverb+adjective’ combine is ‘x’ and ‘adverb+verb’ combine is ‘y’; then the net sentiment score of these two taken together will be $x + 0.3y$, if the weightage factor for ‘adverb+verb’ combine is 30%.

The implementation version involving ‘adverb+adjective’ combination only is referred hereafter as SWN(AAC). The ‘AAC’ in it is used as short form of ‘adverb+adjective combine’. As stated earlier, we have chosen a scaling factor $sf = 0.35$, equivalent to giving only 35% weight to ‘adverb’ scores, when ‘adverb’ and ‘adjective’ scores are added up.. The modifications in adjective scores are thus in a fixed proportion to adverb scores. Since we chose a value of scaling factor $sf = 0.35$, the adjective scores will get a higher priority in the combined score. The indicative pseudo-code for this scheme is illustrated below:

```

For each sentence, extract adv+adj
combines.
For each extracted adv+adj combine do:
• If adj score=0, ignore it.
• If adv is affirmative, then
  o If score(adj)>0
    ▪  $fs_{AAC}(adv, adj) = \min(1, score(adj) + sf * score(adv))$ 
  o If score(adj)<0
    ▪  $fs_{AAC}(adv, adj) = \min(1, score(adj) - sf * score(adv))$ 
• If adv is negative, then
  o If score(adj)>0
    ▪  $fs_{AAC}(adv, adj) = \max(-1, score(adj) + sf * score(adv))$ 
  o If score(adj)<0
    ▪  $fs_{AAC}(adv, adj) = \max(-1, score(adj) - sf * score(adv))$ 

```

Here, adj refers to ‘adjectives’ and adv refers to ‘adverbs’. The final sentiment values (fs_{AAC}) are scaled form of adverb and adjective SentiWordNet scores, where the adverb score is given 35% weightage. The presence of ‘not’ is handled by negating the obtained polarity score for a word from the SentiWordNet library. First of all we extract sentence boundaries of a review and then we process all the sentences. For each sentence we extract the adv+adj combines and then compute their sentiment scores as per the scheme de-

scribed in the pseudo-code. The final document sentiment score is then an aggregation of sentiment scores for all sentences occurring in it. The aggregate score value determines the polarity of the review. If the aggregate score is above a threshold value (usually 0), the document is labeled as 'positive' and 'negative' otherwise.

The implementation version involving both 'adverb+adjective' and 'adverb+verb' sentiment scores is referred hereafter as SWN (AAAVC). The 'AAAVC' in this is used as short form of 'adverb+adjective, adverb+verb combine'. It is similar to the previous scheme in its way of combining adverbs with adjectives or verbs, but differs in the sense that it counts both adjectives and verbs for deciding the overall sentiment score. The indicative pseudo-code of key steps for this scheme is illustrated below:

```

For each sentence, extract adv+adj and
adv+verb combines.
1. For each extracted adv+adjcombine do:
  • If adj score=0, ignore it.
  • If adv is affirmative, then
    o If score(adj)>0
      ▪ f(adv,adj)=
        min(1,score(adj)+sf*score(adv))
    o If score(adj)<0
      ▪ f(adv,adj)= min(1,score(adj)-
        sf*score(adv))
  • If adv is negative, then
    o If score(adj)>0
      ▪ f(adv,adj)= max(-
        1,score(adj)+sf*score(adv))
    o If score(adj)<0
      ▪ f(adv,adj)= max(-1,score(adj)-
        sf*score(adv))
2. For each extracted adv+verb combine do:
  ▪ If verb score=0, ignore it.
  ▪ If adv is affirmative, then
    • If score(verb)>0
      o f(adv,verb)=
        min(1,score(verb)+sf*score(adv))
    • If score(verb)<0
      o f(adv,verb)=
        min(1,score(verb)-
        sf*score(adv))
  ▪ If adv is negative, then
    • If score(verb)>0
      o f(adv,verb)= max(-
        1,score(verb)+sf*score(adv))
    • If score(verb)<0
      o f(adv,verb)= max(-
        1,score(verb)-sf*score(adv))
3. fAAAVC(sentence)=f(adv,adj)+
  0.3*f(adv,verb)

```

Since we need to combine 'adverb+adjective' and 'adverb+verb' scores together, we have tried with different aggregation weights for 'adverb+verb' scores with respect to 'adverb+adjective' patterns. No single weight assignment appears to work well for all the six datasets. The figures 4 and 5 show the variation performance with change in the weightage fac-

tor. The occurrence of 'not' has been handled in a similar manner as in previous scheme. The 'adverb+adjective' and 'adverb+verb' polarity in each sentence are then aggregated and the overall aggregated value for the document then decides its polarity. If it is greater than a threshold (usually 0), the document is labeled as 'positive' and 'negative' otherwise. A more detailed discussion on our SentiWordNet based implementations are reported in [26] and [27].

3.5 Performance Measures

We have evaluated performance of four different implementations for sentiment analysis on six different datasets. Our performance evaluation involved computation of standard metrics of Accuracy, Precision, Recall, F-measure and Entropy. The expression for computing Accuracy is:

$$Accuracy = \frac{NOCC}{n} \quad (8)$$

where, NOCC is Number of Correctly Classified Document and n is the total number of documents. The expressions for Precision, recall and F-Measure are as shown in the equations below:

$$Pr(l, c) = \frac{n_{lc}}{n_c} \quad (9)$$

$$Re(l, c) = \frac{n_{lc}}{n_c} \quad (10)$$

$$Fmeasure(l, c) = \frac{2*Re(l, c)*Pr(l, c)}{Re(l, c) + Pr(l, c)} \quad (11)$$

$$Fmeasure = \sum_i \frac{n_i}{n} \max(Fmeasure(i, c)) \quad (12)$$

where, n_{lc} is number of documents with original label l and classified label c ; $Pr(l, c)$ and $Re(l, c)$ are the Precision and Recall respectively; n_c is number of documents classified as c , and n is number of documents in original class with label l . The expression for Entropy E is as per the equation below:

$$E_c = - \sum_l P(l, c) * \log(P(l, c)) \quad (13)$$

$$E = \sum_c \frac{n_c^* E_c}{n} \quad (14)$$

where, $P(l, c)$ is the probability of documents of classified class with label c belonging to original class with label l , and n is total number of documents. The desired values for Accuracy and F-measure are close to 1 and for Entropy in close to 0. As far as time complexity of the algorithms we implemented is concerned, we did not consider it primarily because of two reasons. One that all approaches are linear in time complexity and second that machine learning classifiers involve

training and test phases whereas SentiWordNet based approaches do not require a training phase. In this situation, it may not be appropriate or useful to compare them in terms of time complexity.

4. RESULTS

We have evaluated performance of four different sentiment analysis schemes on six different datasets. Out of the four implementations, two (NB and SVM) are machine learning classifiers, whereas the remaining two (SWN(AAC) and SWN(AAAVC)) are lexicon-based methods. The table 2 below presents the Accuracy, F-measure and Entropy values for these implementations on all six datasets.

Table 2: Performance results of four methods on all the six datasets.

Dataset	Method	Accuracy (%)	F-Measure	Entropy
Dataset 1	NB	88.40	0.880	0.147
	SVM	84.64	0.846	0.186
	SWN (AAC)	64.07	0.638	0.283
	SWN (AAAVC)	64.71	0.645	0.281
Dataset 2	NB	88.80	0.887	0.143
	SVM	88.90	0.888	0.141
	SWN (AAC)	64.75	0.644	0.281
	SWN (AAAVC)	65.10	0.648	0.280
Dataset 3	NB	90.90	0.910	0.121
	SVM	90.20	0.900	0.134
	SWN (AAC)	78.10	0.768	0.216
	SWN (AAAVC)	79.60	0.777	0.213
Dataset 4	NB	79.47	0.800	0.180
	SVM	86.54	0.870	0.164
	SWN (AAC)	62.65	0.643	0.249
	SWN (AAAVC)	62.92	0.645	0.247
Dataset 5	NB	81.16	0.814	0.190
	SVM	81.91	0.819	0.188
	SWN (AAC)	63.44	0.638	0.272
	SWN (AAAVC)	64.56	0.649	0.268
Dataset 6	NB	83.74	0.836	0.187
	SVM	98.42	0.984	0.135
	SWN (AAC)	56.56	0.559	0.297
	SWN (AAAVC)	58.71	0.562	0.292

The results for accuracy, F-measure and Entropy

indicate that no method is the best across all the datasets. For some datasets NB performs better and for other SVM performs better than NB. The performance level of NB and SVM are close. The SWN(AAC) and SWN(AAAVC) implementations lag behind NB and SVM implementations. The accuracy level for SWN(AAC) varies from 56.56% to 78.10%, whereas for SWN (AAAVC), it varies from 58.71% to 79.60%, across all the six datasets. The SWN(AAAVC) scheme however is relatively superior in performance to SWN(AAC) scheme. The SVM method seems to work best for narrow domain short texts from twitter with accuracy level of more than 98%, whereas SWN approaches perform worst with the twitter dataset. Though machine learning based classifiers perform better than SentiWordNet based approaches, they require prior training with labelled data. In case of SentiWordNet based approaches the performance level is a bit poor than the machine learning classifiers, but they can be implemented without any prior requirement of training. Thus if obtaining an indicative sentiment profile is the goal, SentiWordNet based scheme may be used due to its ease of implementation. However, if accuracy is an important issue, a machine learning classifier would be a better method to use for sentiment analysis.

The table 3 presents total percentage of ‘positive’ and ‘negative’ classifications for all the six datasets, whereas the table 4 specifies the exact number of correctly assigned documents across all the different datasets.

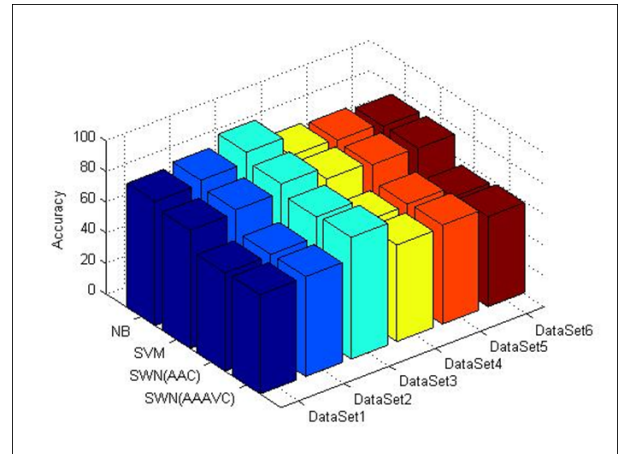


Fig.1: Plot of accuracy values on the six datasets for the four versions implemented.

The figures 1, 2 and 3 present the results for accuracy, F-measure and Entropy, respectively, for the four different methods implemented across six different datasets, in order to have a comprehensive and easy to understand graphical account of the performance of different methods.

We have also tried to find out the best weigh-

Table 3: Total percentage of ‘positive’ and ‘negative’ labels assigned by all four methods.

Dataset	Method	Positive	Negative
Dataset 1	NB	57.57%	42.43%
	SVM	50.92%	49.08%
	SWN(AAC)	58.35%	41.65%
	SWN(AAAVC)	57.42%	42.58%
Dataset 2	NB	53.8%	46.2%
	SVM	49.7%	51.3%
	SWN(AAC)	59.05%	40.95%
	SWN(AAAVC)	58.4%	41.6%
Dataset 3	NB	74.6%	25.4%
	SVM	77.7%	22.3%
	SWN(AAC)	82.8%	17.2%
	SWN(AAAVC)	85.5%	14.5%
Dataset 4	NB	43.34%	56.66%
	SVM	29.41%	70.59%
	SWN(AAC)	46.29%	53.71%
	SWN(AAAVC)	47.77%	52.23%
Dataset 5	NB	51.3%	48.7%
	SVM	40.89%	59.11%
	SWN(AAC)	53.40%	46.60%
	SWN(AAAVC)	55.26%	44.74%
Dataset 6	NB	58.48%	41.52%
	SVM	50.1%	49.9%
	SWN(AAC)	38.31%	61.69%
	SWN(AAAVC)	73.75%	26.25%

Table 4: Total number of correctly assigned ‘positive’ and ‘negative’ labels by all four methods.

Dataset	Method	Positive	Negative
Dataset 1	NB	672/700	566/700
	SVM	599/700	586/700
	SWN(AAC)	507/700	390/700
	SWN(AAAVC)	505/700	401/700
Dataset 2	NB	966/1000	810/1000
	SVM	886/1000	892/1000
	SWN(AAC)	738/1000	557/1000
	SWN(AAAVC)	735/1000	567/1000
Dataset 3	NB	709/763	200/237
	SVM	721/763	181/237
	SWN(AAC)	686/763	95/237
	SWN(AAAVC)	704/763	89/237
Dataset 4	NB	387/435	794/1051
	SVM	336/435	950/1051
	SWN(AAC)	284/435	647/1051
	SWN(AAAVC)	297/435	638/1051
Dataset 5	NB	290/318	365/489
	SVM	251/318	410/489
	SWN(AAC)	227/318	285/489
	SWN(AAAVC)	239/318	282/489

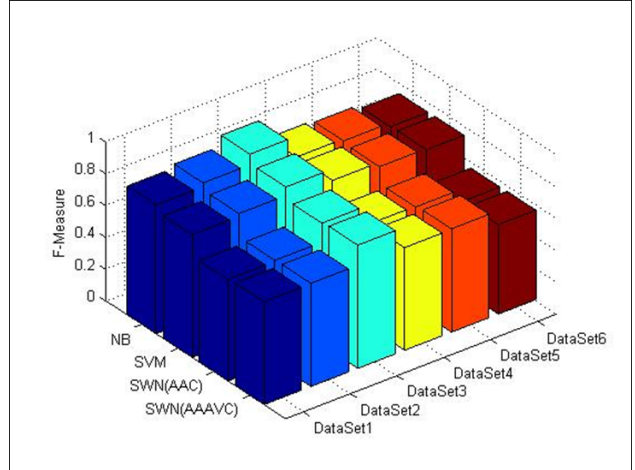


Fig.2: Plot of F-measure values on the six datasets for the four versions implemented.

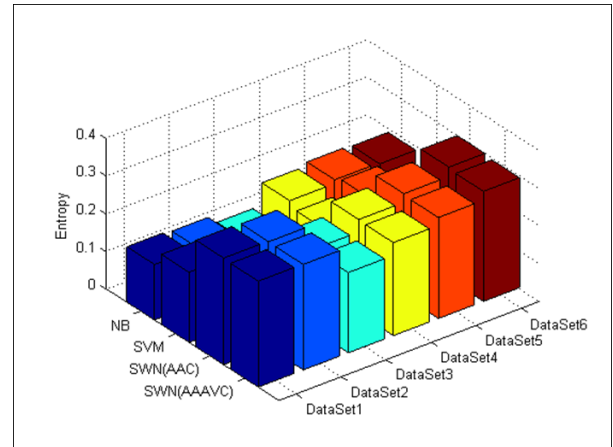


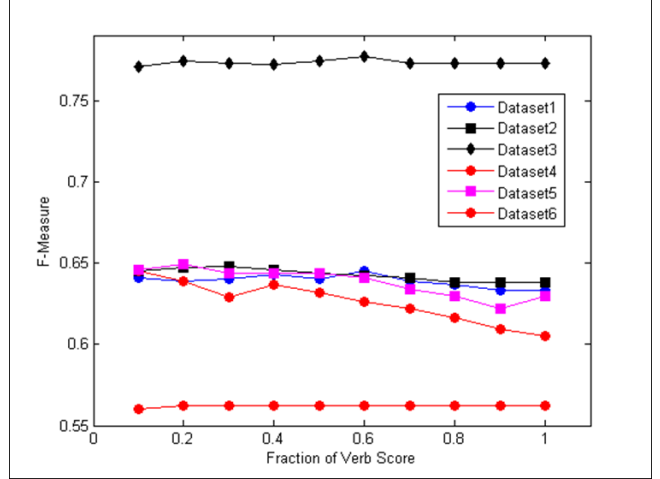
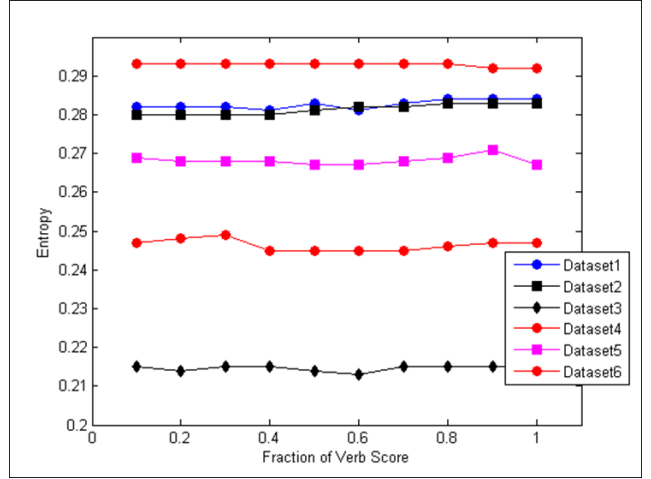
Fig.3: Plot of Entropy values on the six datasets for the four versions implemented.

tag assignment for ‘adverb+verb’ patterns with ‘adverb+adjective’ patterns used in the SWN(AAAVC) scheme. However, there appear to be variation in the best weightage factor across the six datasets. For some dataset, 60% weightage factor gives the best result and for some other 30%. What is however clearly seen is that the net effect of different weightage factor assignment on the performance of SWN(AAAVC) is not very significant. Interestingly, the performance level does not vary a lot on a particular dataset with different weightage factor selection. The figures 4 and 5 present the effect of varying weightage assignment of ‘adverb+verb’ scores with ‘adverb+adjective’ scores, on the F-measure and Entropy levels, respectively. The table 5 presents a detailed account of numerical performance values obtained on these variations.

In summary, we have obtained performance evaluation results for machine learning based classifiers and lexicon based implementation for sentiment anal-

Table 5: Performance of SWN(AA AVC) vis-à-vis different adverb+verb weight assignment.

Dataset		Accuracy	F-Measure	Entropy
Dataset 1	10	64.35	0.641	0.282
	20	64.21	0.639	0.282
	30	64.28	0.640	0.282
	40	64.57	0.643	0.281
	50	64.21	0.640	0.283
	60	64.71	0.645	0.281
	70	64.14	0.639	0.283
	80	63.86	0.637	0.284
	90	63.5	0.633	0.284
	100	63.5	0.633	0.284
Dataset 2	10	64.85	0.645	0.280
	20	65.05	0.647	0.280
	30	65.1	0.648	0.280
	40	64.9	0.646	0.280
	50	64.7	0.644	0.281
	60	64.45	0.642	0.282
	70	64.35	0.641	0.282
	80	64.1	0.638	0.283
	90	64	0.638	0.283
	100	64	0.638	0.283
Dataset 3	10	78.6	0.771	0.215
	20	78.9	0.774	0.214
	30	78.8	0.773	0.215
	40	78.8	0.772	0.215
	50	79.2	0.774	0.214
	60	79.6	0.777	0.213
	70	79.3	0.773	0.215
	80	79.3	0.773	0.215
	90	79.3	0.773	0.215
	100	79.4	0.773	0.215
Dataset 4	10	62.92	0.645	0.247
	20	62.18	0.639	0.248
	30	61.17	0.629	0.249
	40	61.98	0.637	0.245
	50	61.50	0.632	0.245
	60	60.90	0.626	0.245
	70	60.49	0.622	0.245
Dataset		Accuracy	F-Measure	Entropy
	80	59.95	0.616	0.246
	90	59.35	0.609	0.247
	100	58.88	0.605	0.247
Dataset 5	10	64.31	0.646	0.269
	20	64.56	0.649	0.268
	30	64.18	0.644	0.268
	40	64.18	0.644	0.268
	50	64.18	0.644	0.267
	60	63.94	0.641	0.267
	70	63.32	0.634	0.268
	80	62.94	0.630	0.269
	90	62.20	0.622	0.271
	100	63.07	0.630	0.267
Dataset 6	10	58.43	0.56	0.293
	20	58.52	0.562	0.293
	30	58.56	0.562	0.293
	40	58.59	0.562	0.293
	50	58.63	0.562	0.293
	60	58.62	0.562	0.293
	70	58.66	0.562	0.293
	80	58.66	0.562	0.293
	90	58.71	0.562	0.292
	100	58.71	0.562	0.292

**Fig.4:** Variation of F-Measure value with different weightage factors for adverb+verb scores.**Fig.5:** Variation of Entropy values with different weightage factors for adverb+verb Scores.

ysis on six different datasets of different variety. The algorithms are evaluated for their capability of document-level sentiment analysis. It remains to be seen whether the same level of performance will be observed for sentiment analysis at sentence and aspect-levels. It is quite clear from the results that machine learning based classifiers outperform lexicon based methods implemented for document-level sentiment analysis task. However, the requirement of labelled training data, which may not be readily available, is a major problem in applying machine learning classifiers for sentiment analysis. The SentiWordNet based implementations have this advantage but that comes at the cost of reduced accuracy levels. This is the reason why we have explored the applicability and performance of machine learning classifiers and unsupervised lexicon based methods for sentiment analysis. Moreover, the superior performance of machine learning classifiers is not that significant when we ap-

proach to do sentiment analysis at an aspect-level.

5. ASPECT-LEVEL SENTIMENT ANALYSIS

The document-level sentiment analysis is a reasonable measure of positivity or negativity expressed in a review. However, in selected domains it may be a good idea to explore the sentiment of the reviewer about various aspects of the item in that domain, expressed in that review. Moreover, the assumption that a review is only about a single object may not always hold as practically most of the reviews have opinion about different aspects of an item, some of which may be positive while some other negative. For example, a review about a mobile phone may state both positive and negative aspects about certain features of the mobile phone. It may, therefore, be inappropriate to insist on a document-level sentiment polarity expressed in a review for the item. The document-level sentiment analysis is not a complete, suitable and comprehensive measure for detailed analysis of positive and negative aspects of the item under review. The aspect-level sentiment analysis on the other hand allows us to analyze the positive and negative aspects of an item. The aspect-oriented analysis however is often domain specific. It must be known in advance as to which aspects of an item are being opined by the review writers. Once the aspects are identified, opinion targeted about that aspect may be identified and its polarity computed. The aspect-level sentiment analysis thus involves the following: (a) identifying which aspects are to be analyzed, (b) locating the opinionated content about that aspect in the review, and (c) determining the sentiment polarity of views expressed about the aspect.

Due to the domain specific nature of aspect-oriented sentiment analysis, we have chosen to work only on movie reviews. First of all we identified the aspects about movies that are evaluated by the reviewers. For this task, we manually pursued a large number of movie reviews from IMDB and those in our datasets. We also made an elaborate search for identifying the aspects as categorized in different film awards, movie review sites and film magazines. Based on inputs from all these sources, we worked out the list of aspects for which we will compute the sentiment of movie reviews. Since a particular aspect is expressed by different words in different reviews (such as screenplay, screen presence, acting all to refer acting performance), we created an aspect-vector for all aspects under consideration. More precisely, we make a matrix of aspect features, where each row is an aspect and every column in that row contains a synonymous word used by the review and sentiment value on that aspect in each review. The table 6 below presents the indicative structure of the aspect matrix. The synonymous words are stored as comma separated values and Rev. 1 to Rev. N refer to the

N reviews that a movie may have.

Table 6: *Aspect Matrix Structure.*

Aspects Extracted	Synonymous Words	Rev. 1	...	Rev. N
Awards	$V_{1,1}$...	$V_{N,1}$
Editing	$V_{2,1}$...	$V_{N,1}$
Dialog	$V_{3,1}$...	$V_{N,1}$
Cinematography	$V_{4,1}$...	$V_{N,1}$
Choreography	$V_{5,1}$...	$V_{N,1}$
Script/ Story	$V_{6,1}$...	$V_{N,1}$
Music	$V_{7,1}$...	$V_{N,1}$
Filmmaking	$V_{8,1}$...	$V_{N,1}$
Acting/ Performance	$V_{9,1}$...	$V_{N,1}$
Movie Genre	$V_{10,1}$...	$V_{N,1}$
General Comment	$V_{11,1}$...	$V_{N,1}$

After creating aspect vector, we had to locate the opinion about the aspects. In order to this, we parse each review text sentence-by-sentence. First of all we locate any term belonging to aspect vector in the review text. If a sentence contains a mention about an aspect, we select the sentence for sentiment polarity computation about that aspect. It should be mentioned here that sometimes we encounter sentences like “the screenplay is good but the storyline is poor”. For these sentences, it would be difficult to use a simple sentence based sentiment classifier. Therefore, we break these sentences into two, one sentence for each aspect described. Once we get individual sentences, we simply apply the SWN(AAFC) approach for sentiment polarity computation of that aspect. Thus we process all the aspects in one review. This is done for all the reviews of a movie and scores for a particular aspect from all the reviews are combined to have a net sentiment score for that aspect for that movie. Thus, at the end we obtain a sentiment profile of a movie on certain selected aspects.

A summary of steps followed in the aspect-level sentiment profile generation for a movie is given below. The steps are indicative steps for parsing all the reviews of a particular movie. The final sentiment profile of the movie on different aspects is generated based on the aggregation of the aspect-level sentiment result obtained for each movie review. The analysis is now aspect-wise, where we look for opinion about an aspect in all the reviews and this process is repeated for all the aspects under consideration. Different reviews may have different sentiment polarities associated with an aspect. Therefore, all the polarity scores obtained using the SentiWordNet library are aggregated together to have an overall sentiment score for that aspect.

1. Prepare the aspect matrix
2. For each movie
 - Select a review
 - Extract the sentences from review text
 - For each sentence in review
 - a. Locate an aspect
 - b. Compute the sentiment around that aspect using SWN(AA AVC).
3. Aggregate the scores on an aspect from all reviews
4. Generate aggregated score on all aspects of the movie
5. Generate sentiment profile of the movie.

5.1 Result of Aspect-Level Sentiment Classification

We have implemented the aspect-level sentiment analysis work on the movie review dataset 3. For each movie, we scan all its 10 reviews for selected aspects. Thus for n reviews and m aspects, the total scans would be $n \times m$. The sentiment polarities of the desired aspects are computed using SWN(AA AVC) scheme. We present here example results for two selected movies from the dataset 3 on 11 different aspects, including one on the movie in general. The general comment about the movie is usually found in the initial or last sentences of a review. The figures 6 and 7 present sentiment profiles of two different movies, an aspect-level sentiment analysis result. We also display the document-level result for the corresponding movie in the figures to correlate the aspect-level sentiment analysis result with the document-level result.

As seen in figure 6, the sentiment profile is more positively oriented with many aspects rated more positive. This is also congruent to the actual and SWN(AA AVC) obtained document-level result. Similarly, the figure 7 presents the sentiment profile for a movie which appears more negatively oriented in terms of review polarity. The document-level results here (both actual and those obtained using SWN(AA AVC)), have a majority of the reviews negative. This aspect-level result is also congruent to the document-level sentiment analysis result. This method of aspect-level sentiment analysis is thus at least as accurate as the document-level sentiment analysis results. In fact the accuracy levels here may be better than the document-level sentiment analysis result, which however need to be confirmed with more experimental work. The aspect-level sentiment analysis scheme we devised is a very simple lexicon-based method with accuracy levels equivalent to the document-level results. Further, the pictorial representation of sentiment about different aspects of the movie is more expressive and useful than a simple document-level sentiment analysis result. The

method is an unsupervised one and does not require any training data. In fact it can be applied in any domain with the only change required being the aspect matrix. Another, important point to observe is that SWN(AA AVC) seems to work better at aspect-level sentiment analysis task as compared to the document-level sentiment analysis. We need to evaluate this aspect-level sentiment analysis work with some related past work reported in [28]. This would further strengthen our belief in performance level of our aspect-level sentiment analysis implementation.

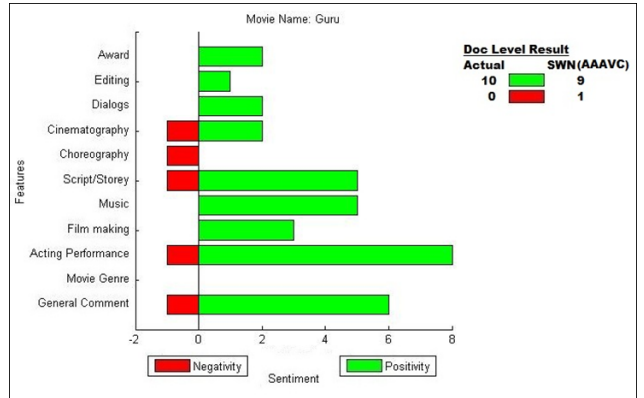


Fig. 6: Sentiment profile of a positively rated movie with actual and observed document-level Sentiment scores.

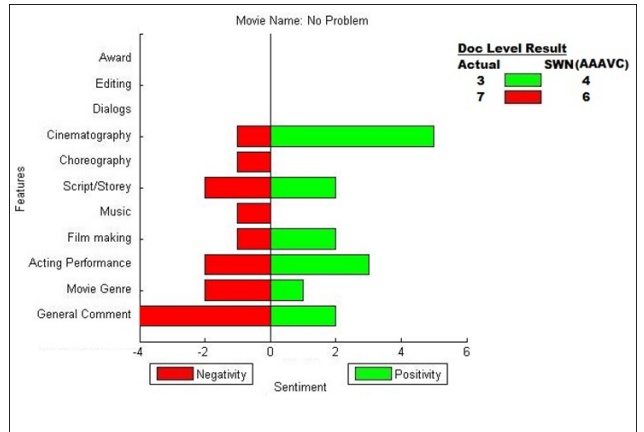


Fig. 7: Sentiment Profile of a negatively rated movie with actual and observed document-level Sentiment scores.

6. CONCLUSIONS

We have done experimental work on performance evaluation of some popular sentiment analysis techniques (both supervised machine learning classifiers and unsupervised lexicon-based). The performance comparison is done at document-level sentiment analysis task. The results demonstrate that machine

learning classifiers obtain better accuracy levels (and better values for other performance evaluation metrics). This is congruent to the earlier findings reported in earlier papers in isolated works. Here, we have made a comprehensive performance evaluation on six datasets, of three different kinds. The techniques are evaluated on movie reviews, blog posts and twitter datasets. The machine learning classifiers however do not seem to be a suitable method for applying on an aspect-based sentiment analysis task. One of the prominent reasons for this is lack of availability of labeled training data. Moreover, the difference in accuracy levels of machine learning classification and unsupervised lexicon-based approaches seem to diminish in aspect-level sentiment analysis work. This is clearly evident from the close congruence of the generated sentiment profile for movies and their actual document-level sentiment labels. This shows that lexicon-based methods are not inherently inferior in performance. What actually goes against achieving better accuracy levels in document-level sentiment analysis task with lexicon-based methods is that the assumption of review being only about a particular aspect does not hold in practical situations. We have presented a detailed account of both document-level and aspect-levels sentiment analysis task and techniques.

Our experimental work makes three important contributions to the work on sentiment analysis. First, it presents a detailed evaluative account of machine learning and lexicon-based (unsupervised SentiWordNet) sentiment analysis approaches on different kinds of textual data. Second, it explores in depth the use of 'adverb+verb' combine with 'adverb+adjective' combine for document-level sentiment analysis, including the effect of different weightage factor assignments for these scores. Third, it proposes a new and simple aspect-based heuristic scheme for aspect-level sentiment analysis in the movie review domain. The proposed approach results in a more useful sentiment profile for movies and have accuracy levels equivalent to the document-level approach. Moreover, the algorithmic formulation used for aspect-level sentiment profile generation is very simple, quick to implement, fast in producing results and does not require any previous training. It can be used on the run and produces very useful and detailed sentiment profile of a movie on different aspects of interest. This part of the implementation can also be used as an add-on step in movie recommendation systems that use content-filtering, collaborative-filtering or hybrid approaches. The sentiment profile can be used as an additional filtering step for designing appropriate movie recommender systems as suggested earlier in [29] and [30].

References

- [1] B. Liu, "Sentiment analysis and opinion mining," Proceedings of 5th Text Analytics Summit, Boston, June 2009.
- [2] B. Pang, L. Lee & S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques", *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 79-86, Philadelphia, US, 2002.
- [3] B. Pang & L. Lee, "A Sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts, *Proceedings of the ACL*, 2004.
- [4] B. Pang & L. Lee, "Seeing stars: Exploiting class relationship for sentiment categorization with respect to rating scales," *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics*, USA, pp. 115-124, 2005.
- [5] M. Gamon, "Sentiment classification on customer feedback data: Noisy data, large feature vectors and the role of linguistic analysis," *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, pp. 841-847, 2004.
- [6] K. Dave, S. Lawrence & D. Pennock, "Mining the peanut gallery- Opinion extraction and semantic classification of product reviews," *Proceedings of the 12th International World Wide Web Conference*, pp. 519-528, 2003.
- [7] S.M. Kim & E. Hovy, "Determining sentiment of opinions," *Proceedings of the COLING Conference*, Geneva, 2004.
- [8] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 417-424, Philadelphia, US, 2002.
- [9] P. Turney & M.L. Littman, "Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word corpus," *NRC Publications Archive*, 2002.
- [10] D.M. Bikel & J. Sorensen, "If we want your opinion," *International Conference on Semantic Computing*, 2007.
- [11] K.T. Durant & M.D. Smith, "Mining sentiment classification from political web logs," *Proceedings of WEBKDD'06*, ACM, 2006.
- [12] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, 34(1): 1-47, 2002.
- [13] A. Esuli & F. Sebastiani, "Determining the semantic orientation of terms through gloss analysis," *Proceedings of CIKM-05, 14th ACM International Conference on Information and Knowledge Management*, pp. 617-624, Bremen, DE, 2005.
- [14] R. Prabowo & M. Thelwall, "Sentiment analysis:

- A combined approach,” *Journal of Informetrics*, 3, pp. 143-157, 2009.
- [15] B. Pang & L. Lee, “Opinion mining and sentiment analysis,” *Foundations and Trends in Information Retrieval* 2(1-2), pp. 1-135, 2008.
- [16] C.D. Manning, P. Raghavan & H. Schutze, “*Introduction to Information Retrieval*,” Cambridge University Press, New York, USA, 2008.
- [17] Weka Data Mining Software in JAVA, <http://www.cs.waikato.ac.nz/ml/weka/>
- [18] F. Benamara, C. Cesarano & D. Reforigiato, “Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone,” *Proceedings of ICWSM 2006*, CO USA, 2006.
- [19] M. Karamibeker & A.A. Ghorbani, “Verb oriented sentiment classification,” *Proceedings of International Conference on web Intelligence and Intelligent Agent Technology*, 2012.
- [20] P. Chesley, B. Vincent, L. Xu & R.K. Srihari, “Using verbs and adjectives to automatically classify blog sentiment,” *American Association for Artificial Intelligence*, 2006.
- [21] <http://www.cs.cornell.edu/people/pabo/movie-review-data/>
- [22] Internet Movie Database, <http://www.imdb.com>
- [23] D. Mahata & N. Agarwal, “What does everybody know? Identifying event-specific sources from social media,” *Proceedings of the fourth International Conference on Computational Aspects of Social Networks (CASoN 2012)*, Sao Carlos, Brazil, 2012.
- [24] Alchemy API, retrieved from www.alchemyapi.org on Dec. 15, 2012.
- [25] Twitter Sentiment Analysis dataset, available at <http://www.textanalytics.in/datasets/twittersentim>
- [26] V. K. Singh, R. Piryani, A. Uddin & P. Waila, “Sentiment analysis of movie reviews and blog posts: Evaluating SentiWordNet with different linguistic Features and scoring schemes,” in *Proceedings of 2013 IEEE International Advanced Computing Conference*, Ghaziabad, India, IEEE, Feb. 2013.
- [27] V.K. Singh, R. Piryani, A. Uddin & P. Waila, “Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification,” *Proceedings of the 2013 International Multi-Conference on Automation, Communication, Computing, Control and Compressed Sensing*, Kerala, India, IEEE, pp. 712-717, 2013.
- [28] T.T. Thet, J.C. Na & C.S.G. Khoo, “Aspect-based sentiment analysis of movie reviews on discussion boards,” *Journal of Information Science*, 36(6), pp. 823-848, 2010.
- [29] V. K. Singh, M. Mukherjee & G. K. Mehta, “Combining collaborative filtering and sentiment analysis for improved movie recommendations,” in *C. Sombattheera et al. (Eds.): Multi-disciplinary Trends in Artificial Intelligence*, LNAI 7080, Springer-Verlag, Berlin-Heidelberg, pp. 38-50, 2011.
- [30] V. K. Singh, M. Mukherjee & G. K. Mehta, “Combining a content filtering heuristic and sentiment analysis for movie recommendations,” in *K.R. Venugopal & L.M. Patnaik (Eds.): ICIP 2011, CCIS 157*, pp. 659-664, Springer, Heidelberg, 2011.



Vivek Kumar Singh received Master's and Doctoral degree in Computer Science from the University of Allahabad, Allahabad, India during 2001 and 2010, respectively. From 2004 to 2011, he has been Assistant Professor in Computer Science at Banaras Hindu University, Varanasi, India. Currently he is working as Assistant Professor in Computer Science at South Asian University, New Delhi, India. He is a senior member of IEEE, and member of ACM and IEEE-CS. His research interests include Collective Intelligence and Text Analytics. His research on text Analytics is funded by DST, Govt. of India and UGC, Govt. of India.



Rajesh Piryani obtained Bachelors' degree in Computer Engineering from Tribhuvan University, Kathmandu, Nepal in 2010 and Masters' degree in Computer Application from South Asian University, New Delhi, India in 2013. His research interests include Sentiment Analysis, Information Extraction and Semantic Annotation. Rajesh is a member of IEEE.



Pranav Waila has obtained Masters Degree in Computer Application from Pondicherry Central University, India during 2005-2008. Currently he is Doctoral program student at Banaras Hindu University, Varanasi, India. Earlier he worked in industry sector assignments in SCM Microsystems Chennai, Huawei Technology and MakeMyTrip. His broad research interest lies in computational matchmaking, recommender

systems and social media analytics. Pranav is a student member of ACM and IEEE.



Madhavi Devaraj received Master of Computer Applications and M.Phil. Degrees in Computer Science from Madurai Kamaraj University, Madurai, India in 2000 and 2004, respectively. She is currently a Ph.D. student at Gautam Buddha Technical University, Lucknow, India. Earlier she was an Assistant Professor in Invertis Institute of Management and Technology, Bareilly, India from July 2006 to Feb. 2007 and

in Babu Banarasi Das University, Lucknow, India from April 2012 to April 2013. Her research interests include Algorithmic applications on Information Extraction and Sentiment Analysis.